Xây dựng hàm băm trên các cấp số nhân cyclic

Constructing Hash Function Based on Cyclic Geometric Progressions

Hồ Quang Bửu, Ngô Đức Thiện, Trần Đức Sự

Abstract: *Hash functions have an important role in modern cryptography; they are used in digital signature; authentication... Hash function schemes are constructed on block ciphers. In this paper a new method to implement a hash function with Matyas-Mayer-Oseas scheme is proposed, but the cipher block is constructed based on cyclic geometric progressions. Some estimation about a new hash function is also presented.*

I. MỞ ĐẦU

Cụm từ *hàm băm* có nguồn gốc lịch sử từ khoa học máy tính, nó biểu thị một hàm dùng để nén một chuỗi đầu vào tùy ý thành một chuỗi có độ dài cố định ở đầu ra. Hàm băm còn được phổ biến rộng rãi dưới cái tên *hàm băm mật mã*. Hàm băm sẽ tạo ra một kết quả ở đầu ra từ bản tin đầu vào, đầu ra này được biết đến với nhiều tên khác nhau: mã băm, kết quả băm, giá trị băm, mã xác thực. Hàm băm dùng để tính giá trị băm của một tài liệu số (văn bản số, ảnh số,...). Giá trị băm có thể xem như "đại diện" của tài liệu số hay "tóm lược" thông báo và được sử dụng trong một số ứng dụng như: Xác thực tính toàn vẹn của dữ liệu; xác thực số, chữ ký số, bảo vệ bản quyền tài liệu số, nhận dạng mật khẩu; nhận dạng đối tượng...

1. Định nghĩa hàm băm

Định nghĩa 1: Hàm băm h là một hàm có ít nhất hai tính chất sau:

+ *Tính chất nén:* h sẽ ánh xạ một đầu vào x có độ dài bit hữu hạn tuỳ ý tới một đầu ra $h(x)$ có độ dài bit n hữu hạn.
+ Tính chất dễ dàng tính toán: Với h cho trước và một đầu vào x, có thể dễ dàng tính được $h(x)$.

2. Một số tính chất của hàm băm không khoá

Giả sử h là một hàm băm không có khoá, x và x' là các đầu vào, y và y' là các đầu ra tương ứng. Ngoài hai tính chất cơ bản trên ta còn có 3 tính chất sau:

a) Tính khó tính toán nghịch ảnh:

Đối với hầu hết các đầu ra được xác định trước, khó có khả năng tính toán để tìm một đầu vào bất kỳ mà khi băm sẽ cho ra đầu ra tương ứng (Tức là tìm một nghịch ảnh x' sao cho $h(x') = y$ với y cho trước và không biến đổi đầu vào tương ứng).

b) Khó tìm nghịch ảnh thứ hai:

Khó có khả năng tính toán để tìm một đầu vào đã cho trước (Tức là với x cho trước phải tìm $x' \neq x$ sao cho $h(x) = h(x')$)

c) Tính khó va chạm. Khó có khả năng tính toán để tìm hai đầu vào khác nhau bất kỳ x và x' để sao cho $h(x) = h(x')$.

Định nghĩa 2: Hàm băm một chiều (OWHF - oneway hash function).

OWHF là một hàm băm có tính chất bổ sung là:

- Khó tìm nghịch ảnh
- Khó tìm nghịch ảnh thứ hai.

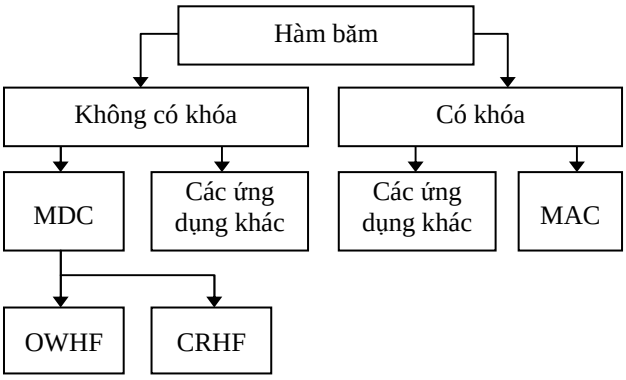
Định nghĩa 3: Hàm băm khó va chạm (CRHF: Collision Resistant HF)

CRHF là một hàm băm có tính chất bổ sung là:

- Khó tìm nghịch ảnh thứ hai
- Khó và chạm

II. CÁC SƠ ĐỒ XÂY DỰNG HÀM BĂM

Phân loại các hàm băm cho trong sơ đồ Hình 1.
Trong đó: *MAC: Message Authentication Code.*
MDC: Modification Detection Code.



Hình 1. Phân loại các hàm băm

1. Các hàm băm không có khoá

Định nghĩa 4: Mật mã khối (n,r) là một mã khối xác định một hàm khả nghịch từ các bản rõ n bit sang các bản n bit bằng cách sử dụng một khoá r bit. Nếu E là một phép mã hoá như vậy thì $E_k(x)$ ký hiệu cho phép mã hoá x bằng khoá k.

Định nghĩa 5: Cho h là một hàm băm có lặp được xây dựng từ một mật mã khối với hàm nén f thực hiện s phép mã hoá khối để xử lý từng khối bản tin n bit. Khi đó tốc độ của h là $1/s$.

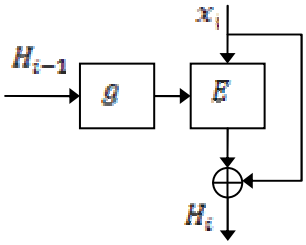
a. MDC độ dài đơn

Ba sơ đồ Hình 2, 3 và 4 dưới đây có liên quan chặt chẽ với các hàm băm độ dài đơn, xây dựng trên các mật mã khối. Các sơ đồ này có sử dụng các thành phần được xác định trước như sau:

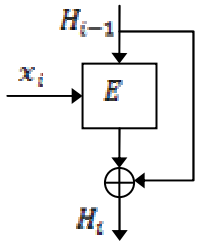
- Một mật mã khối n bit khởi tạo E_k được tham số hoá bằng một khoá đối xứng k.

- Một hàm g ánh xạ n bit vào thành khoá k sử dụng cho E (Nếu các khoá cho E cũng có độ dài n thì g có thể là hàm đồng nhất).

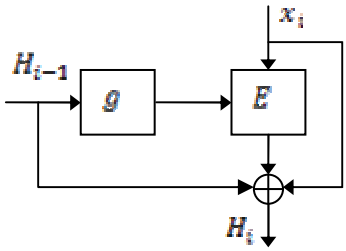
Một giá trị ban đầu IV thích hợp dùng với E.



Hình 2. Matyas-Mayer–Oseas



Hình 3. Davies-Mayer



Hình 4. Sơ đồ Miyaguchi – Preneel

b. MDC độ dài kép: MDC-2 và MDC-4

MDC-2 và MDC-4 là các mã phát hiện sự sửa đổi yêu cầu tương ứng là 2 và 4 phép toán mã hoá khối trên mỗi khối đầu vào hàm băm. Các sơ đồ này sử dụng 2 hoặc 4 phép lặp của sơ đồ M-D-O để tạo ra hàm băm có độ dài kép. Khi sử dụng DES thì MDC-2 và MDC-4 sẽ tạo ra mã băm 128 bit. Tuy nhiên trong cấu trúc tổng quát có thể dùng các hệ mật mã khối khác. MDC-2 và MDC-4 sử dụng các thành phần xác định như sau:

- DES được dùng làm mật mã khối E_k có đầu vào /đầu ra là 64 bit và với khoá k 56 bit.

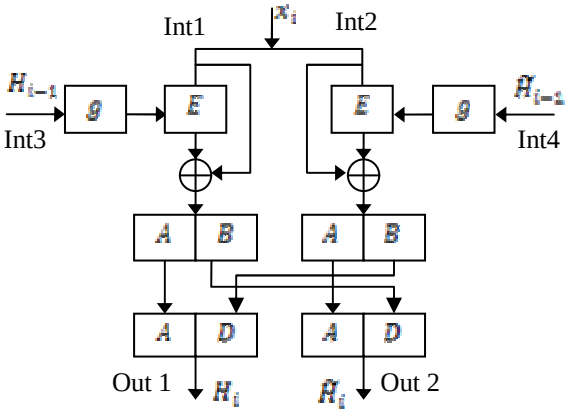
- Hai hàm g và $\tilde{g}$ ánh xạ các giá trị 64 bit U thành các khoá DES 56 bit như sau:

Cho $U = u_1u_2 \dots u_{64}$, xoá mọi bit thứ 8 và đặt các bit thứ 2 và thứ 3 về "10" đối với g, và "01" đối với $\tilde{g}$.

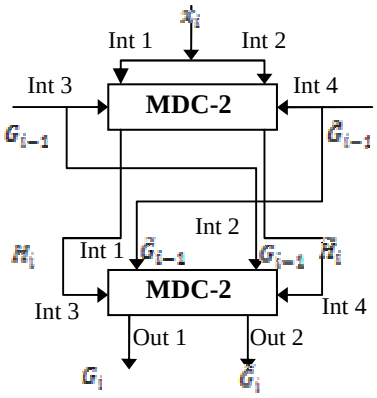
$$g(U) = u_1 10 u_4 u_5 u_6 u_7 u_9 u_{10} \dots u_{63}$$

$$\tilde{g}(U) = u_1 01 u_4 u_5 u_6 u_7 u_9 u_{10} \dots u_{63}$$

Đồng thời phải đảm bảo các khoá DES không yếu hoặc nửa yếu vì các khoá loại này có bit thứ hai bằng bit thứ ba, và cũng đảm bảo yêu cầu bảo mật là $g(IV) \neq \bar{g}(IV)$.



Hình 5. Thuật toán MDC-2



Hình 6. Thuật toán MDC-4

Các thuật toán MDC-2 và MDC-4 được mô tả trong Hình 5 và Hình 6.

2. Các hàm băm có khoá (MAC)

Các hàm băm có khoá được sử dụng để xác thực thông báo và thường được gọi là các thuật toán tạo mã xác thực thông báo (MAC).

MAC dựa trên các mật mã khối, thuật toán như sau:

Vào: Dữ liệu *x*, mật mã khối *E*, khoá MAC bí mật *K* của *E*.

Ra: *n* bit MAC trên *x* (*n* là độ dài khối của *E*)

(1) Độn và chia khối: Độn thêm các bit vào *x* nếu cần. Chia dữ liệu đã độn thành t khối, mỗi khối *n* bit: $x_1, x_2, \ldots, x_t$.

(2) Xử lý theo chế độ CBC.

Ký hiệu E_k là phép mã hoá *E* với khoá *K*.

Tính khối H_t như sau:

$$H_1 \leftarrow E_k(x_1)$$

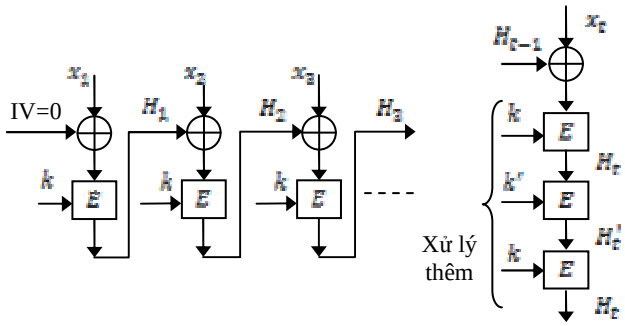
$$H_i \leftarrow K_k(H_{i-1} \oplus x_i) \quad 2 \leq i \leq t$$

(3) Xử lý thêm để tăng sức mạnh của MAC:

Dùng một khoá bí mật thứ hai $K' \neq K$. Tính:

$$H_t' \leftarrow E_{K'}^{-1}(H_t); \quad H_t = E_K(H_t')$$

(4) Kết thúc: MAC là khối n bit H_t.



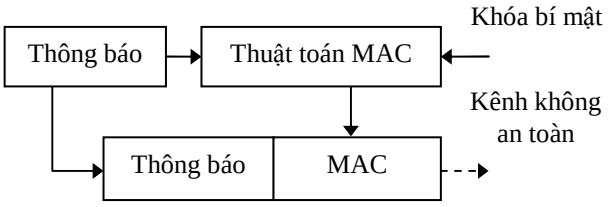
Hình 7. Sơ đồ Miyaguchi – Preneel

3. Một số phương pháp toàn vẹn dữ liệu và xác thực thông báo

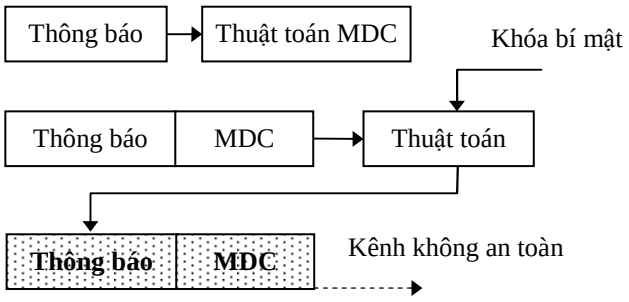
Xác thực thông báo là một thuật ngữ được dùng tương đương với xác thực nguyên gốc của dữ liệu.

Có ba phương pháp xác định tính toàn vẹn của dữ liệu bằng cách dùng các hàm băm:

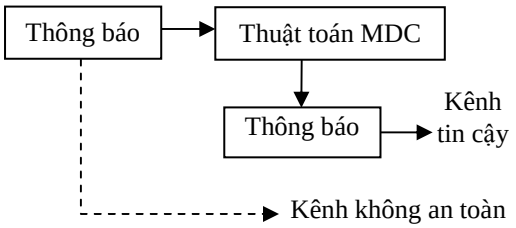
- Chỉ dùng MAC (Hình 8).
- Dùng MDC và mã hoá (Hình 9).
- Sử dụng MDC và kênh tin cậy (Hình 10)



Hình 8. Toàn vẹn dữ liệu dùng MAC



Hình 9. Toàn vẹn dữ liệu dùng MDC và mã hóa



Hình 10. Toàn vẹn dữ liệu dùng MDC và kênh tin cậy

III. XÂY DỰNG HÀM BĂM KHÔNG KHÓA TRÊN CÁC CẤP SỐ NHÂN CYCLIC

1. Nhóm nhân của vành đa thức

Định nghĩa 6: Tập các đa thức $f(x)$ trong vành đa thức $\mathbb{Z}_2[x]/x^n+1$ với một phép toán nhân đa thức tạo nên một nhóm nhân G.

$$\langle f(x);*\rangle = G$$

Nếu $g(x); f(x) \in G$ thì $g(x) * f(x) = d(x) \in G$. Trong nhóm nhân tồn tại phần tử đơn vị $e(x)$ với $f(x) * e(x) = f(x)$.

Bổ đề 1: Trong vành $\mathbb{Z}_2[x]/x^n+1$ với $n = 2^k$, tập các đa thức có trọng số lẻ sẽ tạo nên một nhóm nhân G các đa thức theo modulo x^n+1 [1].

Bổ đề 2: Mọi phần tử trong nhóm nhân *G* có cấp là 2^k hoặc có cấp là ước của 2^k[1].

Bổ đề 3: Số các thặng dư bậc hai trong nhóm nhân *G* của vành được xác định như sau [1]:

$$|Q| = 2^{2^{k-1}-1}$$

Các phần tử cấp n và nhóm nhân cyclic cấp n.

Xét $a(x) \in G$ với $a(x) \in \sum a_i x^i$. Ta có bổ đề sau:

Bổ đề 4: Đa thức $a(x)$ là phần tử cấp n khi nó có chứa một số lẻ các đơn thức có mũ lẻ có cấp n và một số chẵn các đơn thức có mũ chẵn có cấp là ước của n. Số các đa thức cấp *n* bằng 2^{n-2} [1].

Ví dụ 1: Với $n = 8$, có tất cả $2^6 = 64$ các phần tử cấp n. Ta có thể sử dụng các phần tử này để xây dựng các nhóm nhân cyclic cấp n.

$$A_i = \{a_i(x), a_i^2(x), a_i^3(x), \dots, a_i^{n-1}(x), a_i^n(x) = 1\}$$

Có tất cả 2^{n-2} các nhóm nhân cyclic cấp n và nhóm nhân I cũng thuộc vào lớp các nhóm nhân này. Ta gọi I là nhóm nhân cyclic đơn vị.

2. Các cấp số nhân cyclic cấp n

Nếu ta nhân các phần tử của một nhóm nhân cyclic cấp n với một phần tử bất kỳ trong nhóm nhân G của vành đa thức ta sẽ thu được một cấp số nhân có công bội là phần tử sinh của nhóm nhân và có số hạng ban đầu là đa thức đem nhân.

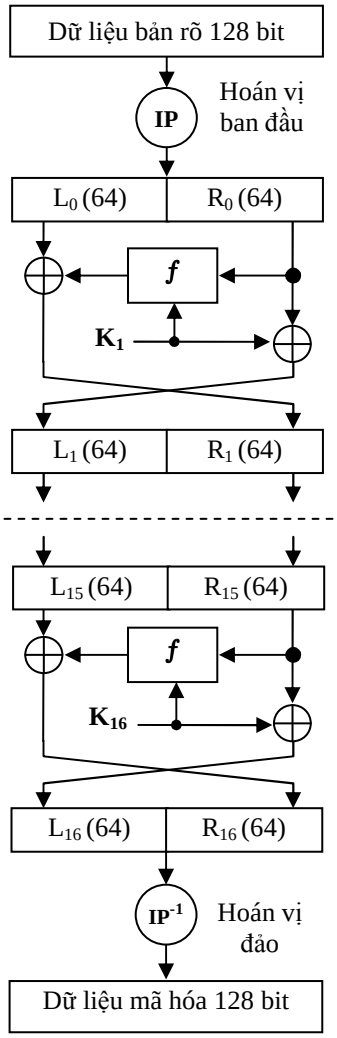
Bổ đề 5: Số các cấp số nhân cyclic cấp n xây dựng được trong *G* được xác định như sau [1]:

$$N = 2^{2^k-1}.2^{2^k-2}$$

3. Xây dựng hàm băm trên các cấp số nhân cyclic

Trong bài báo này chúng tôi đưa ra một phương pháp xây dựng hàm băm 128 bit dựa trên các cấp số nhân cyclic của vành đa thức, với nền tảng là sơ đồ hàm băm Matyas–Mayer–Oseas như Hình 2, đầu ra H_i

có độ dài 128 bit. Khối mật mã E trong sơ đồ này được xây dựng theo mô hình mạng hoán vị thay thế Feistel (Hình 11).



Hình 11. Sơ đồ khối mã hóa E

Khối *E* sẽ mã hóa cho chuỗi bit có độ dài 128. Trong sơ đồ hình 11 các hoán vị IP và hoán vị đảo IP^{-1} được chúng tôi xây dựng và phát triển từ các bảng IP và IP^{-1} của hệ mật DES, cho trong Bảng 1 và Bảng 2.

Hàm f được xây dựng trên cơ sở hệ mật sử dụng các cấp số nhân cyclic trên vành đa thức có hai lớp kề. Các khóa K_i là các phần tử trong một cấp số nhân được chọn như sau [5]:

$$K_i \equiv K_\alpha K_0^i \bmod x^{61} + 1; \quad (i = \overline{1,16})$$

Bảng 1. Bảng hoán vị ban đầu (IP)

122	114	106	98	90	82	74	66	58	50	42	34	26	18	10
124	116	108	100	92	84	76	68	60	52	44	36	28	20	12
126	118	110	102	94	86	78	70	62	54	46	38	30	22	14
128	120	112	104	96	88	80	72	64	56	48	40	32	24	16
121	113	105	97	89	81	73	65	57	49	41	33	25	17	9
123	115	107	99	91	83	75	67	59	51	43	35	27	19	11
125	117	109	101	93	85	77	69	61	53	45	37	29	21	13
127	119	111	103	95	87	79	71	63	55	47	39	31	23	15

Bảng 2. Bảng hoán vị đảo (IP^{-1})

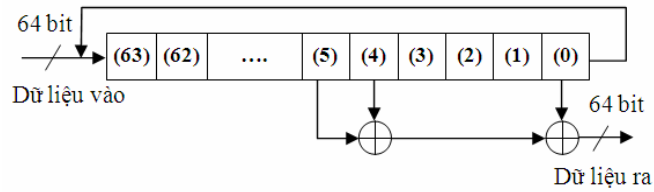
80	16	96	32	112	48	128	64
79	15	95	31	111	47	127	63
78	14	94	30	110	46	126	62
77	13	93	29	109	45	125	61
76	12	92	28	108	44	124	60
75	11	91	27	107	43	123	59
74	10	90	26	106	42	122	58
73	9	89	25	105	41	121	57
72	8	88	24	104	40	120	56
71	7	87	23	103	39	119	55
70	6	86	22	102	38	118	54
69	5	85	21	101	37	117	53
68	4	84	20	100	36	116	52
67	3	83	19	99	35	115	51
66	2	82	18	98	34	114	50
65	1	81	17	97	33	113	49

với K_α là một đa thức có trọng số lẻ tùy ý sao cho: $\deg K_\alpha < 61$; K_0là một phần tử nguyên thủy của nhóm nhân cyclic có cấp bằng $2^{60} - 1$ và cũng là một đa thức có trọng số lẻ [3].

Giả sử ta chọn khóa $K_0 = 1 + x + x^3$,

Phần tử đầu là $K_\alpha = 1 + x + x^2$.

Phần tử đầu của cấp số nhân và cũng là khóa đầu tiên là: $K_1 = K_0 K_\alpha = 1 + x^4 + x^5 \leftrightarrow (045)$. (Chú ý giá trị (045) là dạng biểu diễn số mũ của đa thức). Sơ đồ khối bộ mã hóa f với khóa K_1 như trong Hình 12.



Hình 12. Sơ đồ khối mã hóa f với khóa

$$K_1 = 1 + x + x^3$$

Một khâu mã hóa được thực hiện theo quy tắc:

$$\begin{cases} L_i = R_{i-1} \oplus K_i \\ R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \end{cases}$$

Khối g trong sơ đồ Hình 11 thực hiện việc trích trọn các khóa cho các vòng tiếp theo của quá trình băm. Khối mật mã E trong sơ đồ sử dụng các khóa có độ dài 61 bit. Trong 61 bit khóa ở bước thứ i do khối g tạo ra thì 60 bit đầu tiên sẽ được trích trọn từ 128 bit của H_{i-1} còn bit thứ 61 là bit kiểm tra chẵn lẻ. Việc trích trọn được lấy liên tục các bit cách nhau 2 vị trí trong H_{i-1} (trong khoảng bit 1 đến bit 120). Dưới đây là một vài kết quả đánh giá của hàm băm xây dựng trên các cấp số nhân cyclic.

Bảng 3. Khoảng cách Hamming $d_h(MD_1, MD_i)$ khi các khối dữ liệu khác khối ban đầu 1 bit.

TT	Bản rõ M_i	Giá trị băm MD_i	$d_h(MD_1, MD_i)$
1	0123456789ABCDEF 0123456789ABCDEF	4771249F4AB0E564 908F1456E0D3A239	0
2	**2**123456789ABCDEF 0123456789ABCDEF	7B13337D3B31DC7B 91287CE5FCDFD274	56
3	0**3**23456789ABCDEF 0123456789ABCDEF	995F0EF13134BFAF D43A47B676DFA356	62
4	01**3**3456789ABCDEF 0123456789ABCDEF	00EEA6CB284338F5 704DE9AFEC8A592C	68
5	012**B**456789ABCDEF 0123456789ABCDEF	0BC2CE7B57041014 0609BC377F579110	62
6	0123**0**56789ABCDEF 0123456789ABCDEF	E1EDDCED0686C4F6 E1DACEE0AA906D52	64
7	0123**4**76789ABCDEF 0123456789ABCDEF	09B62BF471BA9644 A9C64A47762FF6BD	60
8	0123**5**7789ABCDEF 0123456789ABCDEF	0CA4992082D77070 73C61EA33CE5D66D	68
9	01234**5**689ABCDEF 0123456789ABCDEF	1E25F66DC86E2888 44CCBDC4367DA463	64
10	012345679**9**ABCDEF 0123456789ABCDEF	1AD3061B585D4602 FBEBA645F50D0203	58
11	012345678**D**ABCDEF 0123456789ABCDEF	E1589BF99823EF04 1210020DA32B4C50	64
12	01234567898BCDEF 0123456789ABCDEF	1EE7BE47AC923862 90929EA7F6E837C1	60
13	0123456789A**F**CDEF 0123456789ABCDEF	BEB0D922F4ECAE48 CF098E0B9CCDF9CE	78
14	0123456789AB**8**DEF 0123456789ABCDEF	AD5C0C8D0A61348B B96861BE92EEBB16	64
15	0123456789ABC**5**EF 0123456789ABCDEF	7906EF1395B4DE95 522E47E70DD5C738	64
16	0123456789ABCD**F**F 0123456789ABCDEF	FD4109489863FD3B 4E79C434BF8355DC	72
17	0123456789ABCDE**B** 0123456789ABCDEF	C6DBEA49E116BEDC 1FF11DF8F7A44A3F	68
18	0123456789ABCDEF **8**123456789ABCDEF	BBE4AE6094334B90 49D253F55195427D	70
19	0123456789ABCDEF 0**3**23456789ABCDEF	BA79EFB1AF077CB5 1988B7580AEA44C1	68
20	0123456789ABCDEF 01**3**3456789ABCDEF	22C2135FCD25DB6C BB0CE7ED5F43BEFE	66
21	0123456789ABCDEF 012**1**456789ABCDEF	ADFA46A0CEC37C5A E0C53DAF31B45B8D	68
22	0123456789ABCDEF 0123**0**56789ABCDEF	A0A88D98F147A0D7 C4284C7EAF58BC1F	68
23	0123456789ABCDEF 01234**1**6789ABCDEF	8DD5B3218D448641 313E52AD01747037	68
24	0123456789ABCDEF 01234**5**7789ABCDEF	62F9919F4FF1A2AE 27C31BD6042FBB04	54
25	0123456789ABCDEF 0123456**5**89ABCDEF	FF3D7A429626EF4E C61B8CF1325300F4	60
26	0123456789ABCDEF 0123456**9**9ABCDEF	48CBABA51460CEF1 4ABFA6A62B4C006B	66
27	0123456789ABCDEF 012345678**D**ABCDEF	A69350AB67BBCC6F 0053037523D9343F	54
28	0123456789ABCDEF 01234567892BCDEF	36F0DCFCCF106D0F 76F938F7FBFBBE0C	56
29	0123456789ABCDEF 0123456789A**A**CDEF	4D16387FD0FA8E8A E12F2ED638A059FF	62
30	0123456789ABCDEF 0123456789AB**4**DEF	422DDC211E659AB0 C3D7A66C29F82331	62
31	0123456789ABCDEF 0123456789ABC**C**EF	8353E1BF4DB4264B 6D7007E1DFA73B51	64
32	0123456789ABCDEF 0123456789ABCD**F**F	7945A131C04B6182 ED6E54D50BE28723	62
33	0123456789ABCDEF 0123456789ABCDE**D**	5DA15DE212D18181 4813623B0E06F874	68

(Chú ý: Trong Bảng 3 và Bảng 4, các ký tự hexa in đậm chứa bit thay đổi).

Bảng 3 là kết quả tính toán phân bố của 32 hàm băm khi thay đổi duy nhất một bit dữ liệu trong 32 khối bản tin rõ so với bản tin ban đầu [2], để thuận tiện cho việc quan sát chúng tôi chỉ thay đổi 1 bit trong chuỗi bản tin đầu tiên của một khối.

Mỗi khối bản tin bao gồm 10 bản tin, mỗi bản tin có độ dài 128 bit. Các hàm băm sử dụng cùng một bộ khóa K khởi tạo:

Phần tử sinh của khóa khởi tạo là đa thức:

$$K_i = 1 + x^9 + x^{18} + x^{27} + x^{36}$$

Phần tử đầu là: $K_a = 1 + x^{10} + x^{20}$.

Phần tử đầu của cấp số nhân (khóa đầu tiên) cũng là khóa khởi tạo sẽ là:

$$K_{IV} = K_1 = K_i K_a = 1 + x^9 + x^{10} + x^{18} + x^{20} + x^{27} + x^{28} + x^{29} + x^{36} + x^{37} + x^{38} + x^{46} + x^{47} + x^{56}$$

Khối bản tin đầu tiên được xây dựng như sau:

Bản tin đầu tiên gồm 32 ký tự dạng hexa (tương ứng 128 bit) được chọn là:

M_1=0123456789ABCDEF0123456789ABCDEF

Các bản tin tiếp theo (từ 2 đến 10) được tạo một cách ngẫu nhiên.

Tiến hành thay đổi lần lượt từng bit từ bit 1 đến bit 128 của bản tin đầu vào M_1, tính khoảng cách Hamming $d_H(MD_1, MD_i)$ của từng lần thay đổi, cuối cùng tính được khoảng cách Hamming trung bình giữa các giá trị băm với giá trị băm ban đầu là:

$$d_{H(tb)} = \frac{1}{128}\sum_{i=1}^{128} d_H(MD_1, MD_i) = 63{,}97$$

Bảng 4 là kết quả tính toán phân bố của bộ mã khi thay đổi khóa khởi tạo K, mỗi khóa khác với khóa đầu tiên 2 bit. Sở dĩ ta phải thay đổi 2 bit (tương ứng thay đổi 2 vị trí) là để đảm bảo đa thức sinh của khóa có trọng số lẻ.

Bản tin đầu vào gồm 10 khối 128 bit được tạo ngẫu nhiên [2].

Chú ý, chiều dài của khóa là 61 bit, do đó khi mô tả khóa bằng 16 ký tự hexa nhưng thực tế chỉ có 15 ký tự đầu là dạng hexa, còn ký tự cuối cùng chỉ có 1 bit nên nó nhận giá trị "1" hoặc "0".

Chọn phần tử đầu của cấp số nhân tạo khóa là:

$$K_a = 1 + x + x^2.$$

Bảng 4. Khoảng cách Hamming $d_H(MD_1, MD_i)$ giữa các cặp giá trị băm khi các khóa khác khóa K_1 2 bit.

TT	Khóa K_i	Giá trị băm MD_i	$d_h(MD_1, MD_i)$
1	123456789ABCDEF1	53C51349140008F9 AA66F954C307AD44	0
2	**B**23456789ABCDEF1	537140BB7C26F26E DD57CFDDA9CE1B8F	64
3	1**7**3456789ABCDEF1	2BA0259D1F16C021 F3A22319AF753ED0	60
4	12**6**456789ABCDEF1	A9FD04E4E1BAC7C0 6119B3FBD8FFD12D	78
5	123**E**56789ABCDEF1	773979064BE2FC31 F0BE347B1EB2D776	72
6	1234**F**6789ABCDEF1	CD24285FFA002E86 5E8AECFACEAB37A5	66
7	12345**C**789ABCDEF1	12F25C6397752342 98EFF42CB48F44A8	64
8	123456**2**89ABCDEF1	764025970C5F0A26 A623D1A24B6D1809	60
9	1234567**D**9ABCDEF1	530804E85FA92A29 C9D3B064481D81F4	52
10	12345678**0**ABCDEF1	36188474DCE9230F 7BFE8799EC1221C4	68
11	123456789**F**BCDEF1	633497CBED502E08 B33AB54809D2DBE2	58
12	123456789A**E**CDEF1	6756EEEBC53E948F E408A13DFF72AA20	66
13	123456789AB**6**DEF1	8778B1FBDE80A5DA 4BEF05156D968B48	58
14	123456789ABC**7**EF1	18DA5D34CA879807 D99ECDBC169ED8AD	74
15	123456789ABCD**B**F1	F9787D06F99822CB 41E264158FF93D0C	64
16	123456789ABCDE**A**1	27395F8741475A8A A3845BB4FB6D0D0E	58

Phần tử sinh khóa đầu tiên K_1 là:

$$K_{1(hex)} = 123456789ABCDEF1 \ \ (W(K_1) = 33).$$

Các khóa K_i chỉ thay đổi 2 bit trong một số hexa của khóa đầu tiên K_1.

Chú ý: vị trí các bit "1" trong các khóa K_i tương ứng là số mũ của x trong đa thức sinh tạo khóa. Ví dụ:

$$K_1 = (123 \ldots F1)_{hex} \leftrightarrow (1000.01000 \ldots . 1111.1)_{bin}$$

$$\leftrightarrow 1 + x^5 + \cdots + x^{56} + x^{57} + x^{58} + x^{59} + x^{60}$$

Khoảng cách Hamming trung bình của các giá trị băm với giá trị băm ban đầu:

$$d_{H(tb)} = \frac{1}{15}\sum_{i=2}^{16} d_H(MD_1, MD_i) = 64{,}13$$

Để khảo sát thêm tính khuếch tán của hàm băm, ta thay đổi cả bản tin rõ và khóa như sau: Giữa nguyên bản tin và lần lượt thay đổi từng bit của khóa K từ bit 1 đến bit 60, bit 61 dùng để kiểm tra chẵn lẻ. Sau đó tính khoảng cách Hamming trung bình giữa các giá trị băm với giá trị băm ban đầu theo công thức:

$$d_{H(tb)} = \frac{1}{60}\sum_{i=2}^{61} d_H(MD_1, MD_i)$$

Thực hiện lặp lại với 10 bản tin được tạo ngẫu nhiên khác nhau ta có kết quả như Bảng 5.

Bảng 5. Tính khoảng cách Hamming trung bình khi thay đổi khóa K và thay đổi bản tin rõ

Bản tin thứ i	*1*	*2*	*3*	*4*	*5*
$d_{H(tb)_i}$	63,27	63,67	64,23	64,67	64,50
Bản tin thứ i	*6*	*7*	*8*	*9*	*10*
$d_{H(tb)_i}$	62,30	63,87	65,37	64,33	63,97

Khoảng cách Hamming trung bình tính được là:

$$d_{H(tb)} = \frac{1}{10}\sum_{i=1}^{10} d_{H(tb)_i} = 64{,}02$$

Qua các kết quả khảo sát khoảng cách Hamming ở trên ta thấy tính khuếch tán của các hàm băm đề xuất là khá tốt.

Để tăng hiệu quả hàm băm ta có thể sử dụng các sơ đồ hàm băm kép với cách xây dựng tương tự như đã trình bày ở trên.

IV. KẾT LUẬN

Bằng việc sử dụng cấu trúc nhóm nhân và cấp số nhân cyclic trên vành đa thức để tạo khóa, ta có thể xây dựng một mật mã khối trên cơ sở mạng hoán vị Feistel với một số ưu điểm sau: (1) việc tính toán khá đơn giản, các khóa được tạo từ các cấp số nhân cyclic và chúng có thể thực hiện được bằng thuật toán nhân và bình phương; (2) số lượng khóa tìm được rất nhiều đáp ứng yêu cầu ngày càng cao của hệ thống; (3) mạch điện mã hóa và giải mã khá đơn giản được thực hiện bởi các thanh ghi dịch và bộ cộng modul 2.

Hệ mật như đề xuất trong bài báo này có thể được sử dụng để xây dựng các hàm băm với độ khuếch tán tốt dùng cho các dịch vụ xác thực và chữ ký số.

TÀI LIỆU THAM KHẢO

[1]. NGUYỄN BÌNH, *Giáo trình Mật mã học*, Học viện Công nghệ Bưu chính Viễn thông, 2004.

[2]. JEAN-YVES CHOUINARD, *ELG 5373 Secure Communi-cations and Data Encryption*, School of Information Technology and Engineering, University of Ottawa, April 2002.

[3]. NGUYEN BINH, LE DINH THICH, *The oders of polynomials and algorithms for defining order of polynomial over polynomial ring*, VICA-5, Hanoi, Vietnam, 2002.

[4]. NGUYỄN BÌNH, LÊ ĐÌNH THÍCH, *Các mã cyclic hệ thống xây dựng từ các mã cyclic cục bộ trên vành đa thức*. REV'98, Hà Nội, Việt Nam, 1998.

[5]. HỒ QUANG BỬU, TRẦN ĐỨC SỰ, *Constructing Interleaved M-sequences over Polynomial Rings with Two Cyclotomic Cosets*, Tạp chí Khoa học và Công nghệ Quân sự, 2011.

Nhận bài ngày: 14/12/2011

SƠ LƯỢC VỀ TÁC GIẢ

HỒ QUANG BỬU

Sinh năm 1972 tại Huyện Vụ Bản Tỉnh Nam Định.

Tốt nghiệp đại học năm 1995 tại Đại học Bách Khoa Đà Nẵng, thạc sĩ năm 2007 tại Học viện Công nghệ BCVT.

Công tác tại Sở Thông tin Truyền thông tỉnh Quảng Nam từ năm 2005.

Lĩnh vực nghiên cứu hiện nay: Lý thuyết mã hóa, mật mã.

Email: hoquangbuu@gmail.com

NGÔ ĐỨC THIỆN

Sinh năm 1974 tại Điện Biên.

Tốt nghiệp đại học năm 1997 tại Đại học Bách Khoa Hà Nội, Thạc sĩ năm 2003 và bảo vệ luận án Tiến sỹ kỹ thuật năm 2010 tại Học viện Công nghệ BCVT.

Công tác tại Học viện Công nghệ Bưu chính Viễn thông từ năm 1997.

Lĩnh vực nghiên cứu: Lý thuyết thông tin, lý thuyết mã hóa và mật mã.

Email: thienptit@yahoo.com

TRẦN ĐỨC SỰ

Sinh năm 1965 tại Nhân Thắng-Gia Bình-Bắc Ninh.

Tốt nghiệp Đại học Bách Khoa Hà Nội năm 1988 và bảo vệ luận án Tiến sỹ kỹ thuật năm 2004 tại Viện Điện tử – Tin học – Tự động hóa.

Hiện là Giảng viên – Học viện Kỹ thuật Mật mã.

Lĩnh vực nghiện cứu: Lý thuyết thông tin, lý thuyết mã hóa, mật mã.

Email: su_attt@yahoo.com